LA-UR-19-23059
Approved for public release; distribution is unlimited.

Title: LANL ARM Benchmarking Efforts

Author(s): Pritchard, Howard Porter Jr.

Intended for: EMC3 day presentation

Issued: 2019-04-04

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

LANL ARM Benchmarking Efforts

Howard Pritchard (howardp@lanl.gov)
EMC3 Day 4/4/19

LA-UR-19-XXXXX

Covered today

- Overview of LANL ARM benchmarking efforts
- Mini-app performance
- ASC/IC app performance results
- ARM future technologies investigations

LANL ARM Benchmark efforts

- Institutional Computing (IC) benchmarking team
  - focus on open science applications
- HPC Applications Readiness (AR) team
  - focus on ASC application porting/performance
  - ARM future technologies effort
- Resources
  - HPE Apollo 70 (TX2s 64 cores/node, 2/4 HT per core)
  - Cray XC50's (TX2s 56 cores/node, 4 HT per core)
  - simulators

Microbenchmark Results

ThunderX2 comparison with Skylake, Broadwell on Darwin

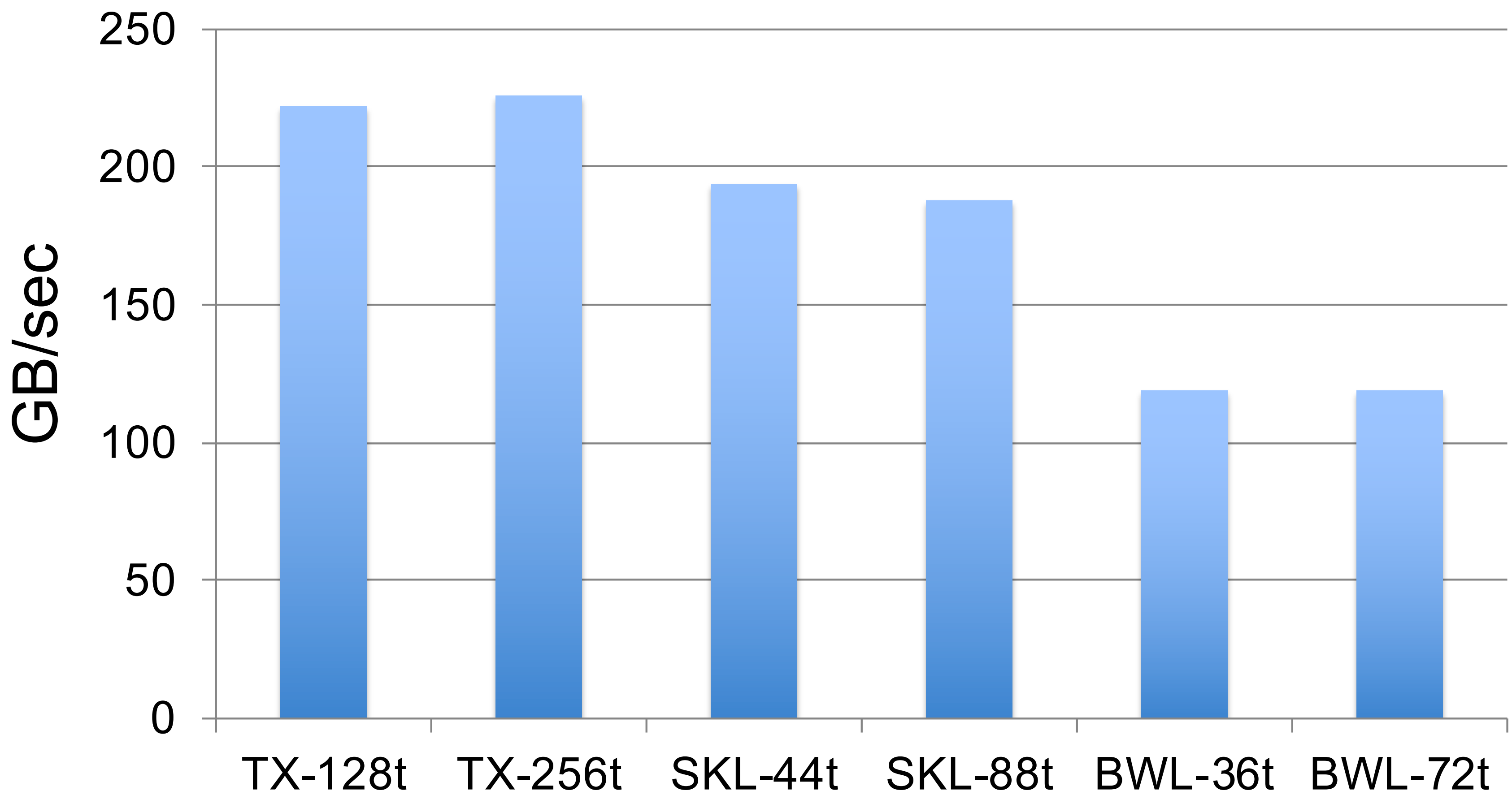
	TX2 (darwin)	BWL	SKL-gold
cores/socket	32 (max 4 HT)	22 (2 HT)	22 (2 HT)
L1 cache/core	32KB I/D (8-way)	32KB I/D (8-way)	32KB I/D (8-way)
L2 cache/core	256KB (8-way)	256 KB (8-way)	1 MB (16-way)
L3 cache/socket	32 MB	33 MB	30.25 MB
#Memory channels/socket	8 DDR4	4 DDR4	6 DDR4
Base clock rate	2.2 GHz	2.2 GHz	2.1 GHz
vector length	128b	256b	512b max

Pagesize 64KB on Darwin ARM nodes

Two ARM partitions – 4tpc and 2tpc configs

Cray ARM systems have 4KB and 2MB pagesizes

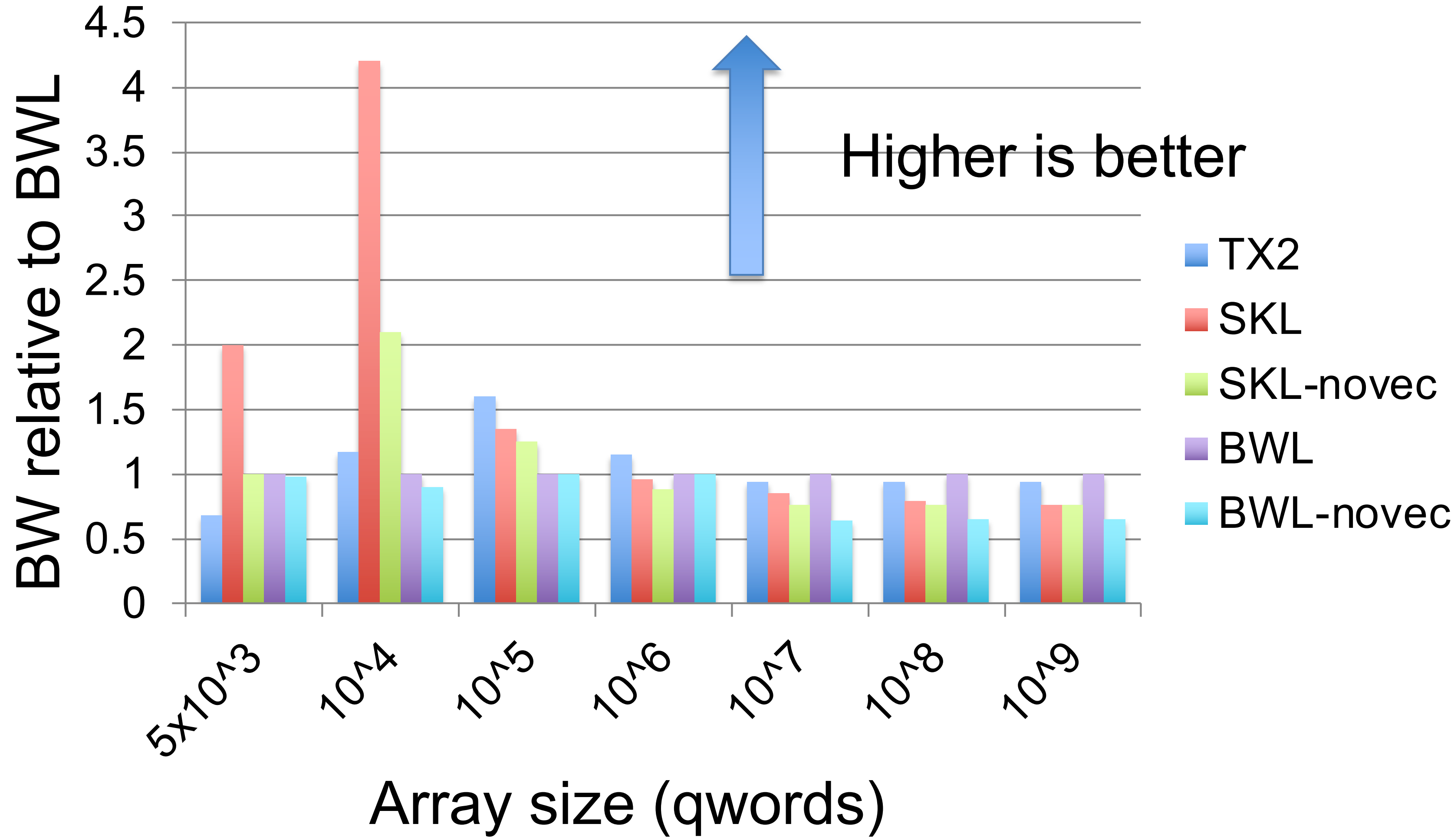
Stream triad benchmark performance comparison – per node basis



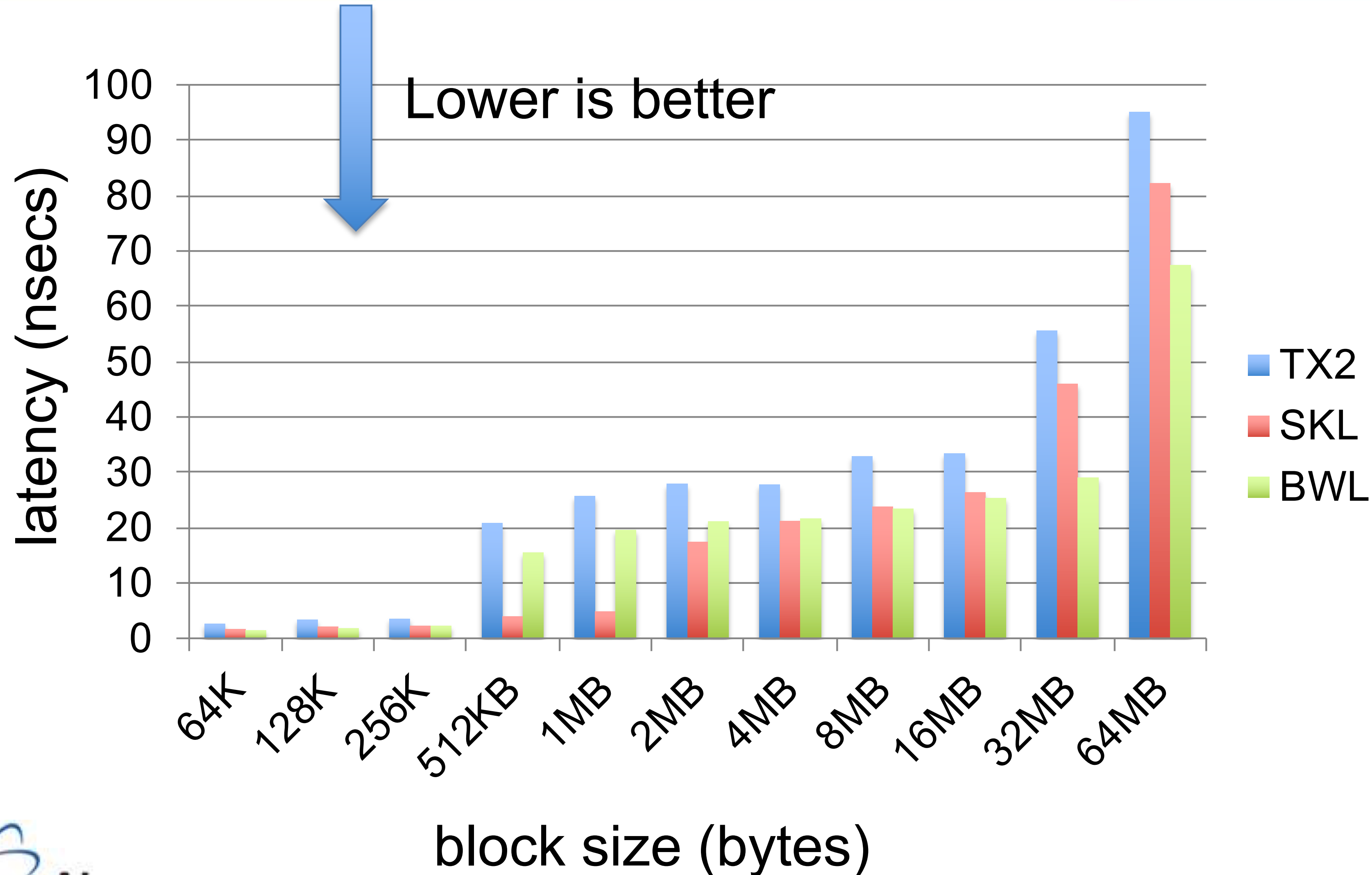
Using 64KB pages on TX2

Note we're off from reported best stream for TX2 by ~20%.

Stream triad benchmark performance comparison – 1 OpenMP thread (normalized to BWL)



Tinybenchmem benchmark performance comparison – dual read test (random accesses)



Mini-Apps Performance Characteristics

Objectives of the mini-app and hopefully later full app performance investigations

- Get a good understanding of current performance limitations of mini-apps on TX2 – focus on single core+HTs and single node performance
- Determine extent to which mini-app kernels can vectorize
- Use Skylake as comparison for scalar and vector performance
- Investigate ability of compilers to generate SVE code for performance critical loops

Mini-apps under consideration

- SNAP – Sn deterministic neutron transport
- Branson – Monte Carlo radiation transport
- Laghos
- xkt (EAP “proxy”)
- RSbench/Xsbench
- HPCG
- Cloverleaf (maybe tealeaf)

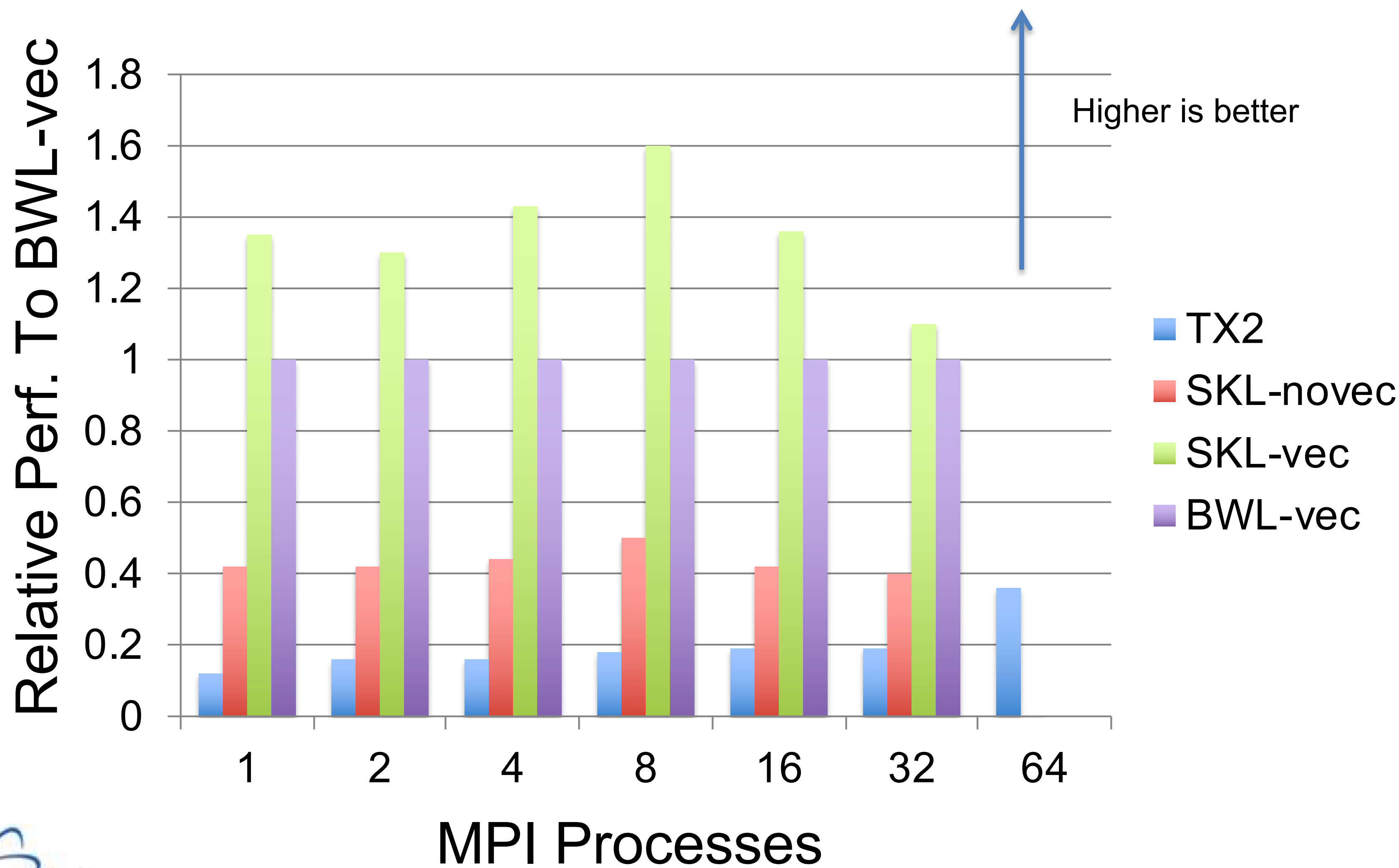
SNAP – Proxy for Partisn

- Sweep algorithm sensitive to performance of caches, memory BW not so important (at least for realistic problems)
- Option to test OpenMP performance
- Crossroads benchmarks
- Sensitive to code generation by compiler for *dim3_sweep,* particularly the fix-up loop. Important for follow-on SVE code generation studies.
- Cray compiler does particularly well here

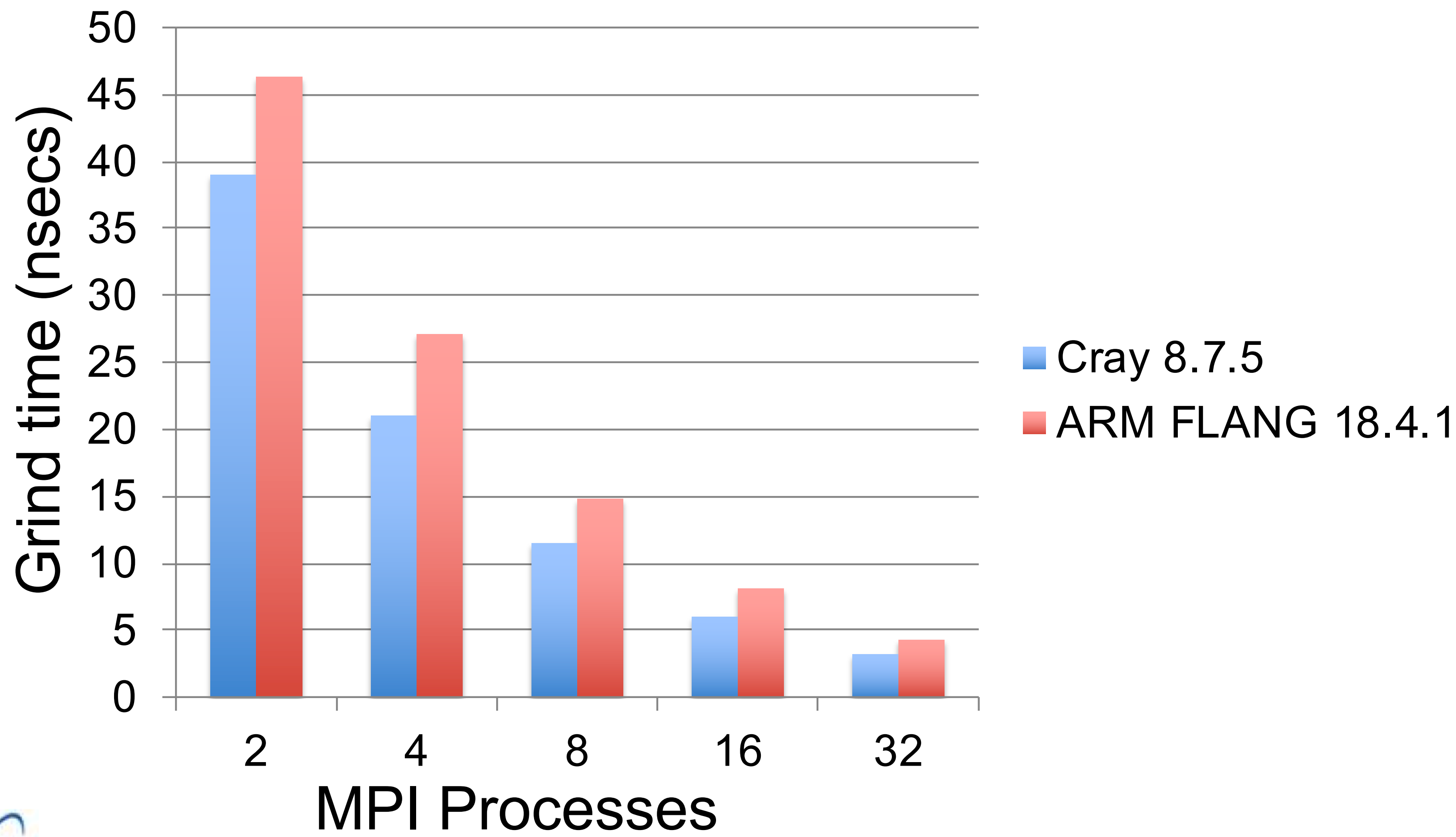
SNAP

- Used Crossroads single node benchmark for runs – inh0001t1 for pure MPI runs
- ARM Flang 18.1.4 and 19.0.0 on ARM nodes, -Ofast
- Intel 18.0.3 with flags in Makefile for isnap. Used skylake-gold for comparison with TX2.
- Used –map-by core option for mpirun
- For hybrid MPI/OpenMP runs also set
  OMP_PLACES=cores
  OMP_PROC_BIND=close

SNAP – Comparison with Skylake-gold and Broadwell – MPI Only – single node (normalized to BWL)



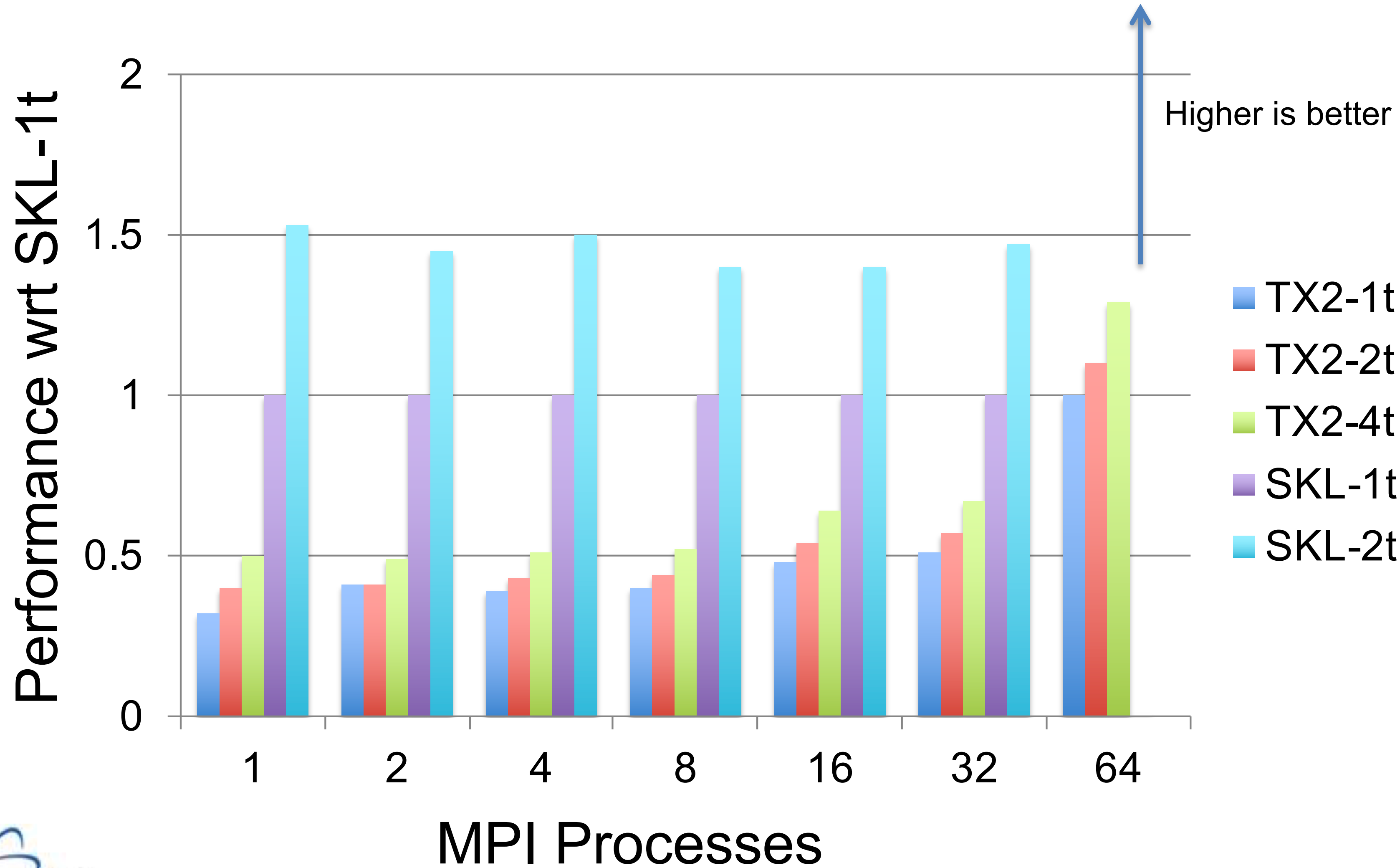
SNAP – CRAY compiler vs ARM FLANG



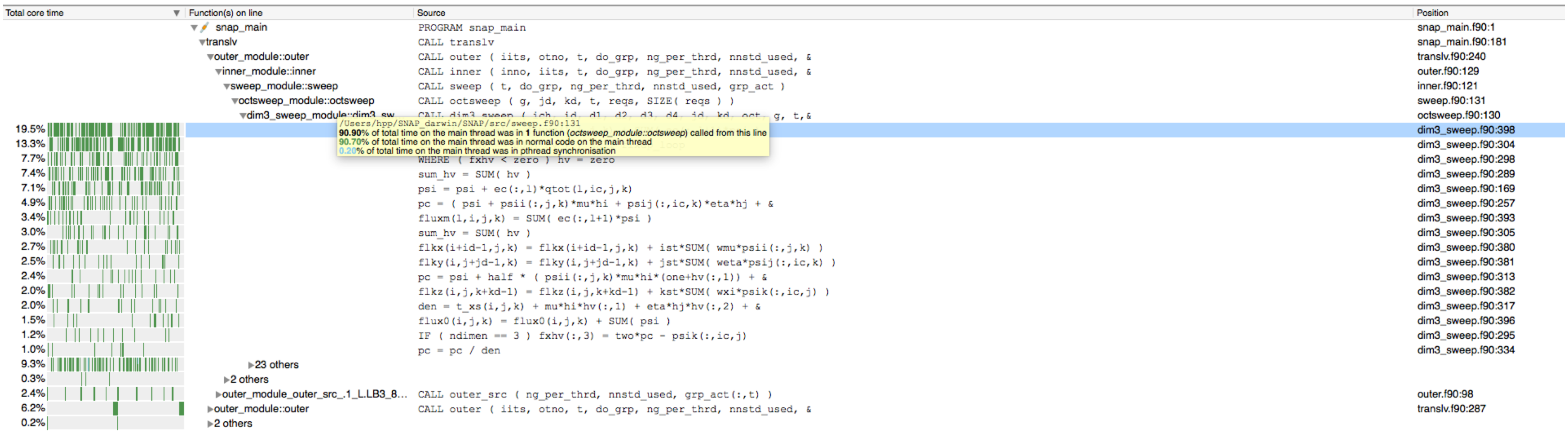
SNAP – OpenMP performance – using those idle HTs

- Used Crossroads inh0001tX (X=1,2,4) single node problem
- Hit turbulence with ARM compilers, ended up using 19.0.0 and using –O3 rather than –Ofast to avoid segfaulting in *inner*
- Hit severe turbulence with Intel compilers, similar segfault traceback but with any optimization level and all available Intel compilers (happened both with/without avx2/avx512 enabled)
- Had success with GNU 8.2.0 on Intel, at the cost of significant loss in base performance

SNAP – OpenMP performance – using those idle HTs (normalized to SKL-1t)



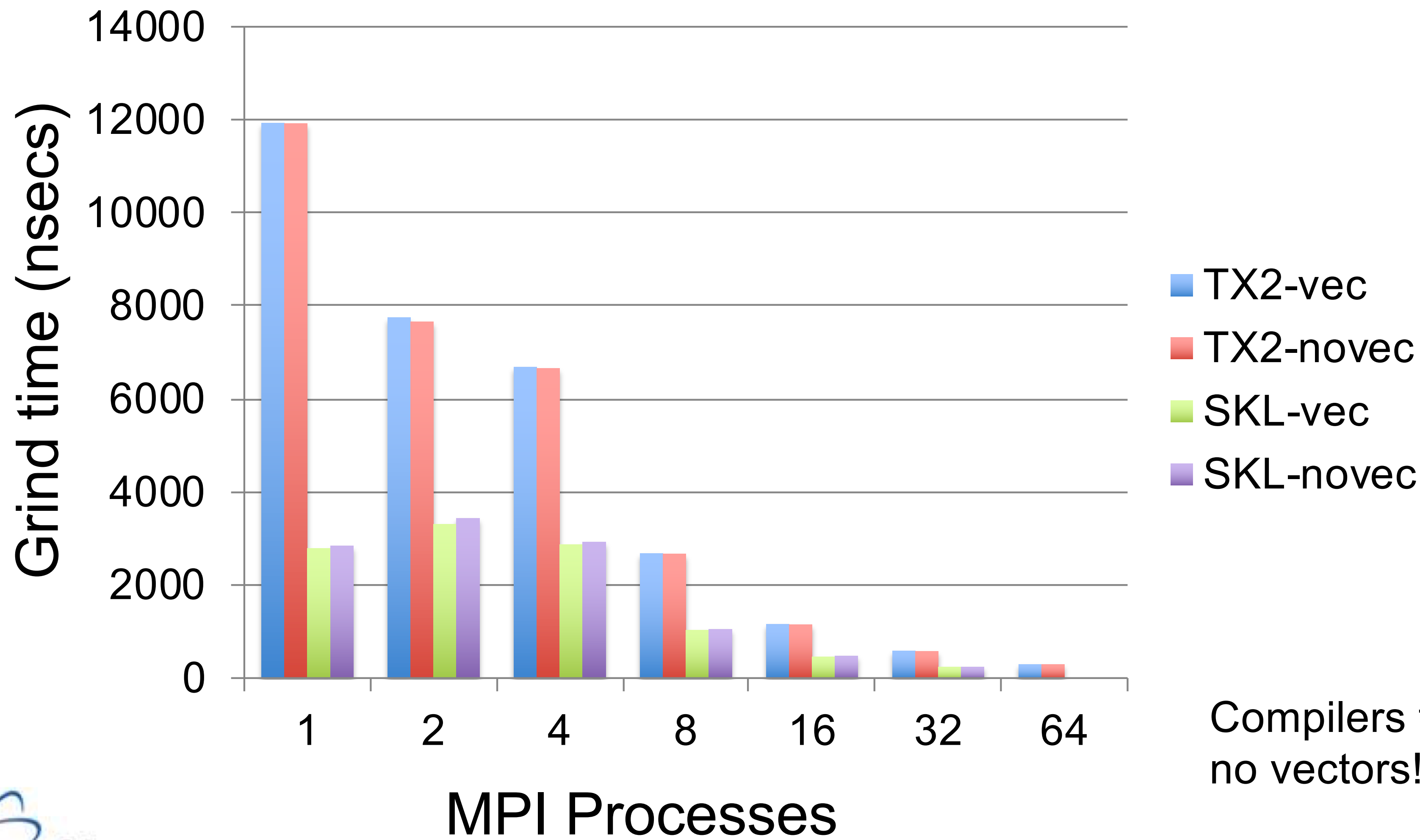
SNAP – MAP profile (small 1core problem)



BRANSON – proxy for Jayenne

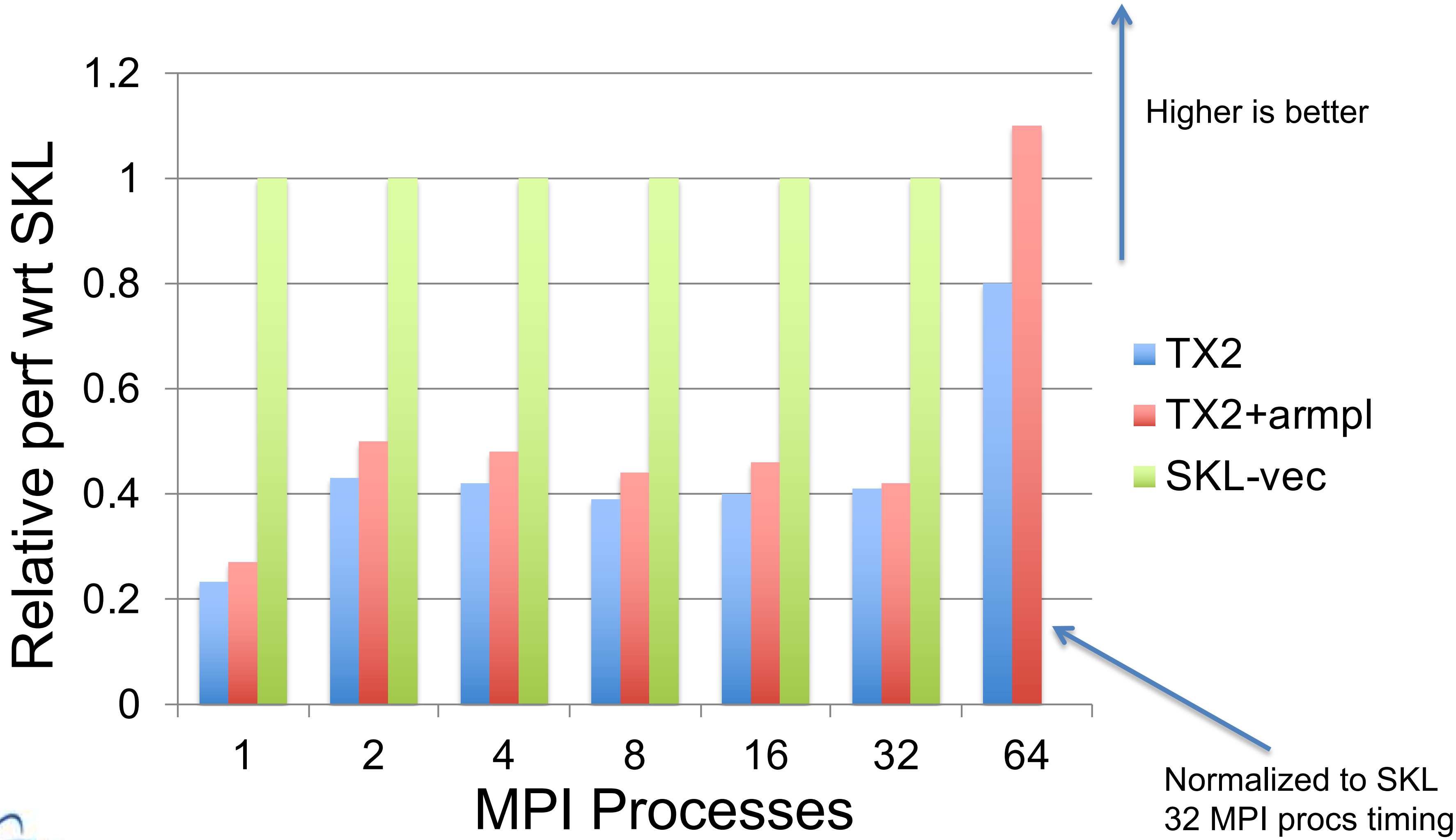
- For proxy_small.xml, appears to be compute bound
- RND123 not yet optimized for ARMv8
- Lots of time spent in math intrinsics (exp, log, sin, cos)
- If this is a good proxy for real app, somewhat disturbing
- Can compilers, using SVE vectorizer find vectors in this code?
- Used proxy_small.xml for runs, NGROUPS=50
- Branson at db66bf52 on GitHub

BRANSON – Skylake-gold vs TX2



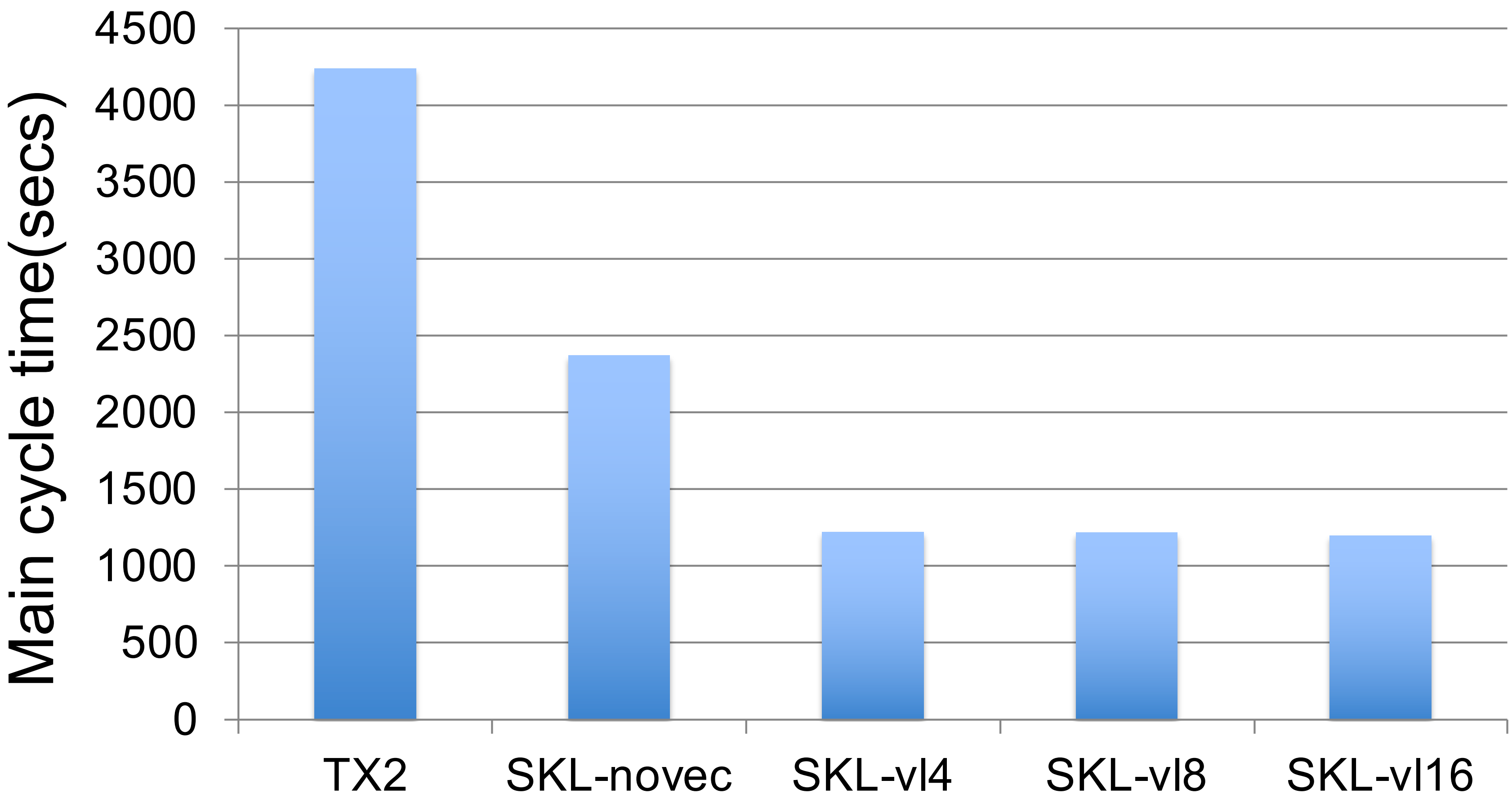
Compilers found no vectors!

BRANSON – Skylake-gold vs TX2 + armpl (normalized to SKL)



ASC/IC TX2/Skylake Application Performance Comparison

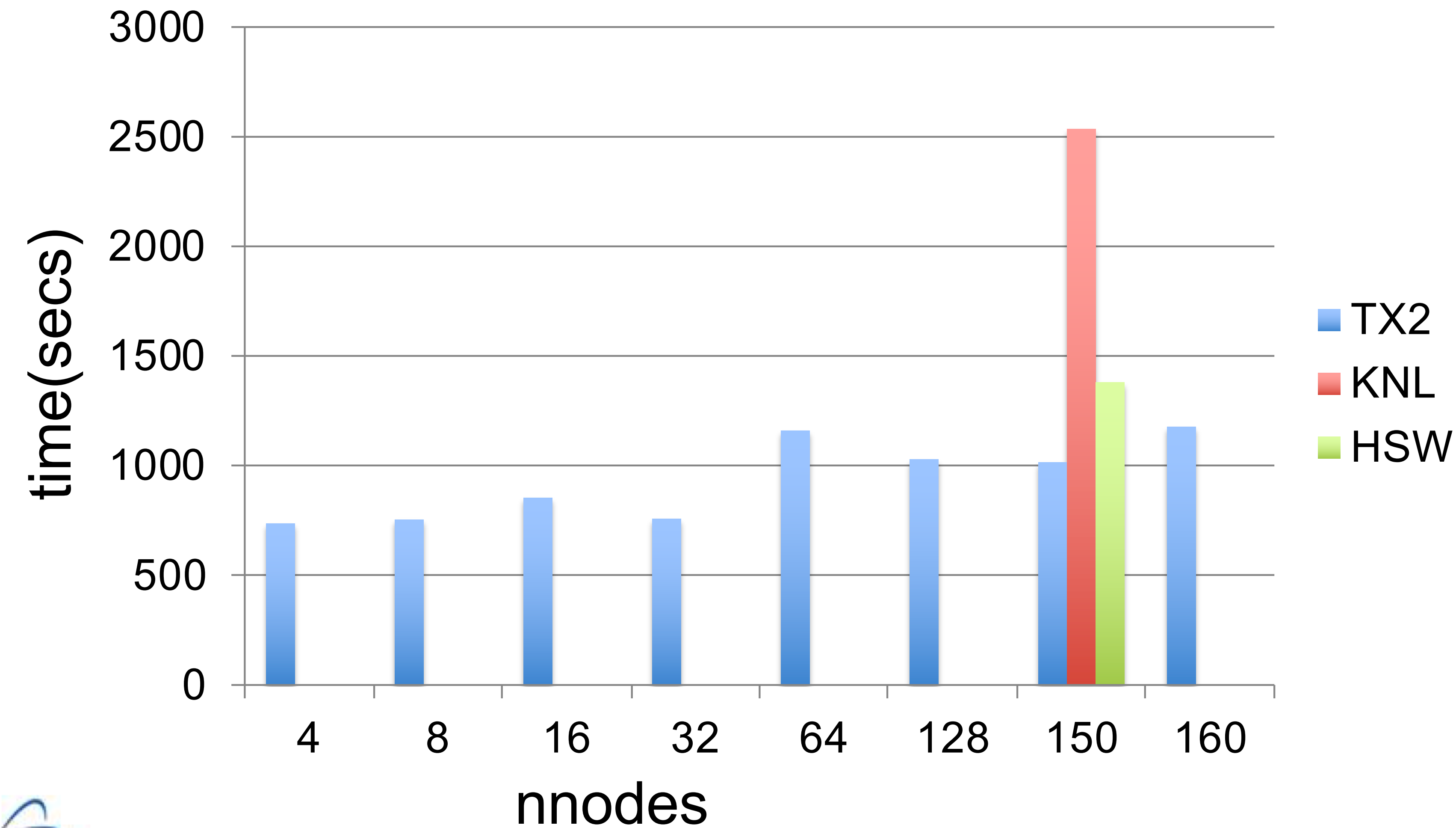
VPIC TX2/Skylake-platinum comparison – per node basis



xRAGE

- Memory BW hungry app that makes ThunderX2 look good
- Weak scaling shaped-charge problem, run on thunder, Cray compilers 8.7.5

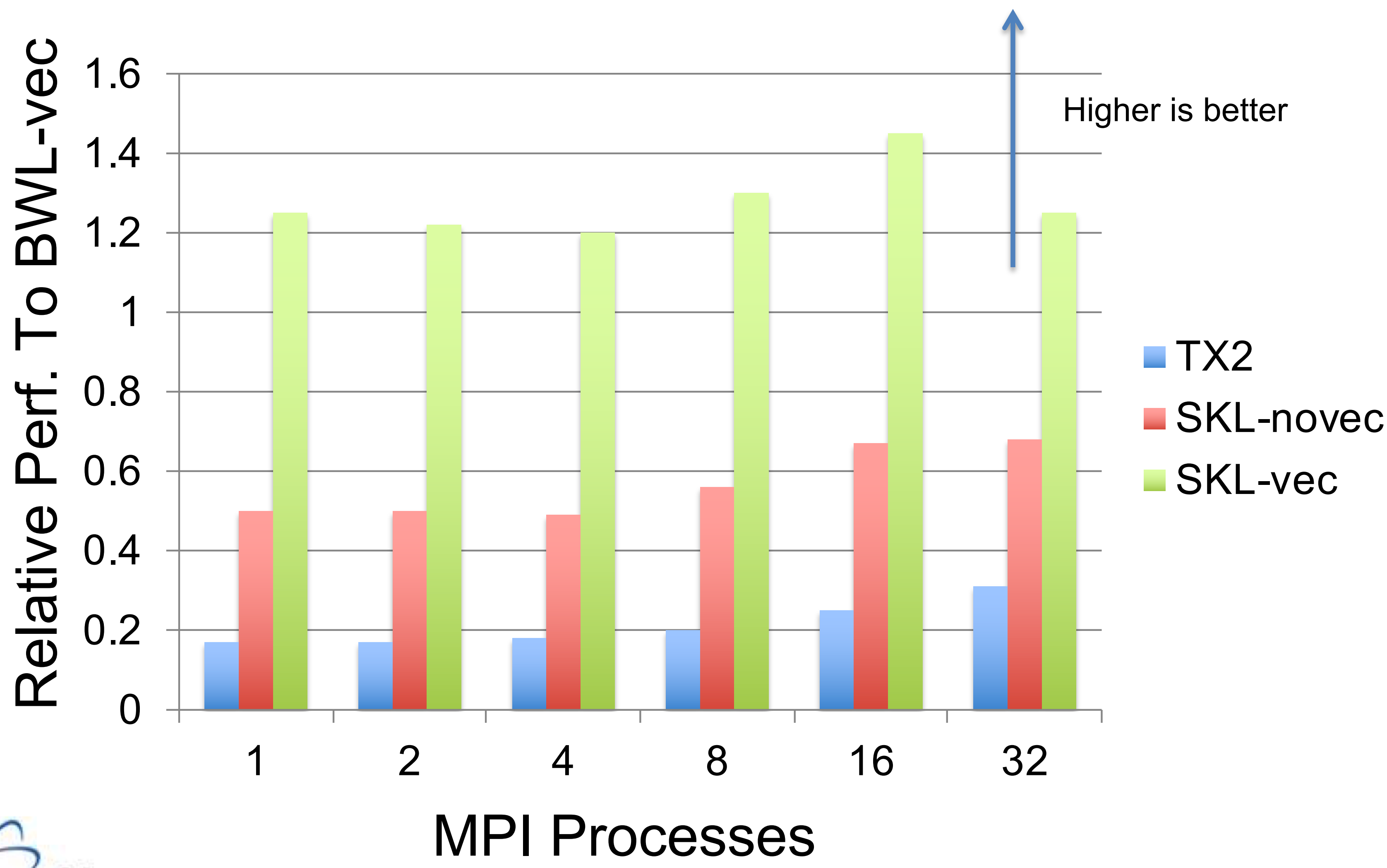
xRAGE shaped charge problem: TX2/KNL/HSW performance comparison



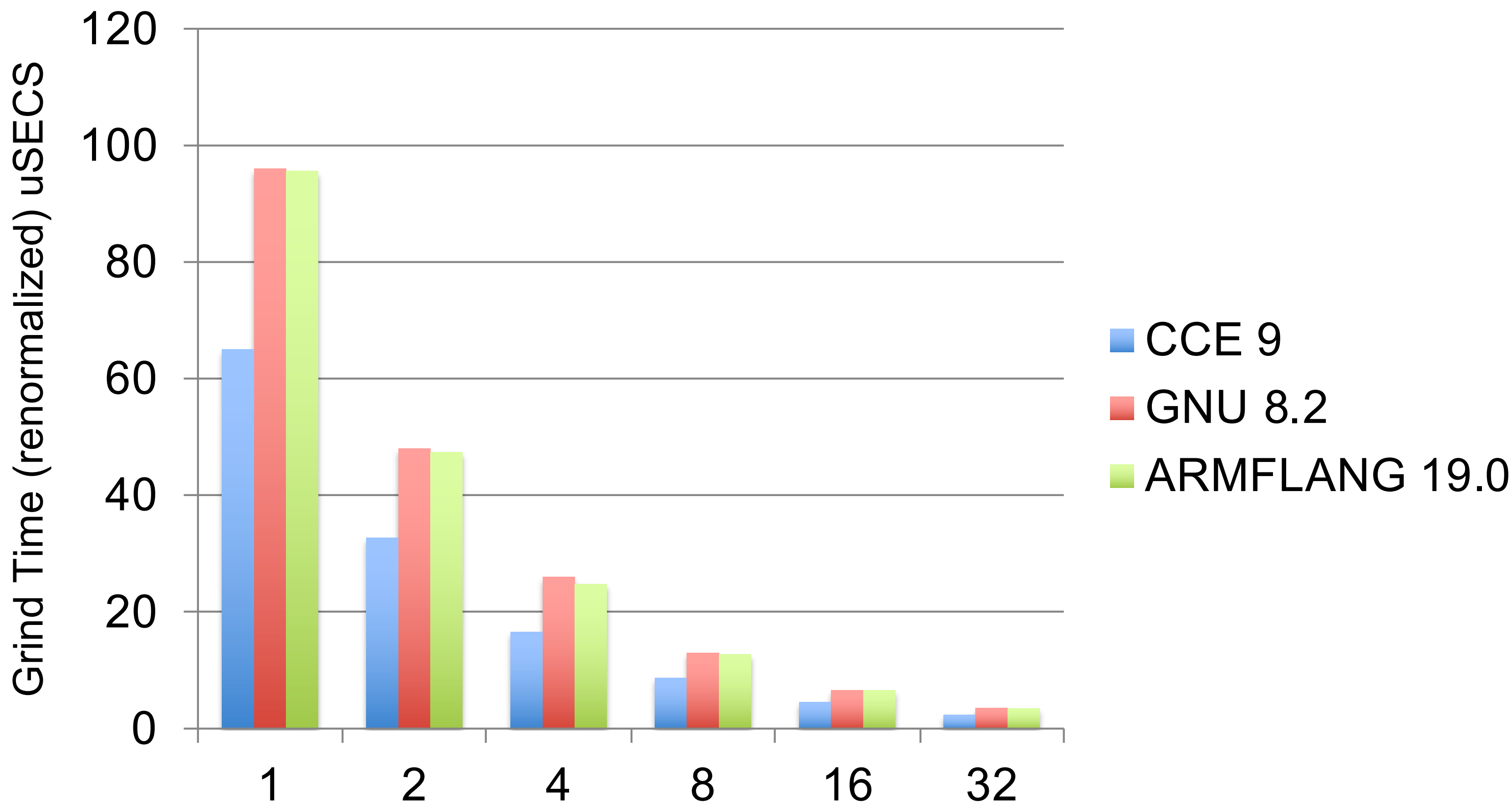
Partisn

- Neutron transport code – deterministic SN method
- Sensitive to cache performance, not typically memory bound
- Vectorizes very well for avx512, NEON

Partisn – Comparison with Skylake-gold and Broadwell – MPI Only – single node (normalized to BWL)



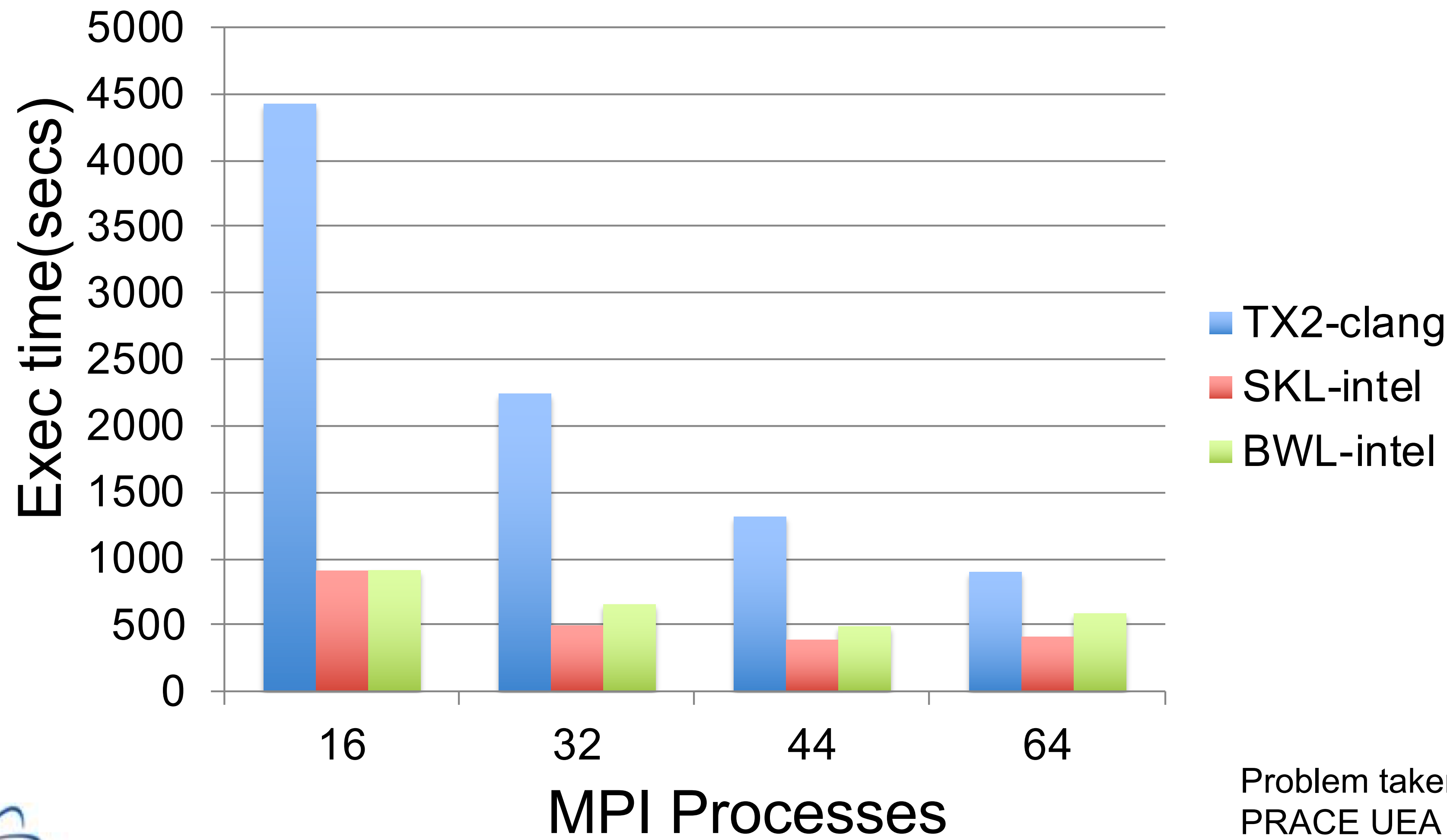
Partisn – Compiler Comparison on Cray XC50



GROMACS

- Uses explicit SIMD instructions via intrinsics. NEON is supported
- Floating point bound (according to GROMACS developers)
- Compilers – Intel 18.0.3 on SKL, GCC 8.2 on ARM

GROMACS: 3.3 Million Atom Problem



Problem taken from PRACE UEA benchmark suite

ARM Future Technologies Effort

Scalable Vector Extension

- New set of SIMD instructions added to the ARMv-8-A instruction set (Hot Chips 2016)
- Vector length agnostic (VLA), supports VL's from 128 to 2048b. For portability, compilers need to generate VLA code
- 32 Z registers, 16 predicate registers, 1 FFR register (first fault register to handle faults in gather/scatter etc.)
- Predicate registers have unique capabilities (not your Cray J90 bitmask registers)
- Fujitsu A64FX processor first with SVE – 512b VL

Exploring SVE

- ARM compilers can generate SVE instructions: -march=armv8-a+sve
- Cray 9.0 - SVE edition compiler can also generate SVE instructions (fixed vector length)
- GNU 8.2 can also generate SVE code
- ARMIE simulator. Available on Darwin. Will use for Marvell/TX4 collab – they want ARMIE traces.
- GEM5 based simulator from ARM, pseudo-A64FX parameter file
- Expect a newer simulator from Marvell soon (good for what if experiments)

External Collaborations

- Sandia Astra program
- Riken/Sandia ARM collaboration – Fiber benchmarks, etc.
- AWE (Tom Deakin, Simon McIntosh-Smith, etc.)
  - simulator effort (they're developing their own, not GEM5)
  - application performance analysis collaboration
- Various vendors

Los Alamos NATIONAL LABORATORY EST. 1943

Operated by Los Alamos National Security, LLC for NNSA